RESEARCH PAPER

Bi-GRU Relation Extraction Model Based on Keywords Attention

Yuanyuan Zhang[1†], Yu Chen[2], Shengkang Yu[1], Xiaoqin Gu[1], Mengqiong Song[1], Yu Peng[1], Jianxia Chen[2] & Qi Liu[2]

[1]Technical Training Center of State Grid Hubei Electric Power Co., Ltd. Wuhan 430070, China

[2]Hubei University of Technology, School of Computer Science, Wuhan 430068, China

Keywords: Relation extraction; Bi-GRU; CRF keywords attention; Hidden similarity

Citation: Zhang, Y.Y. et al.: Bi-GRU Relation Extraction Model Based on Keywords Attention. Data Intelligence 4(3), 552-572 (2022). DOI: 10.1162/dint_a_00147

Receive: Oct. 11, 2021; Revised: Jan. 15, 2022; Accepted: Feb. 10, 2022

ABSTRACT

Relational extraction plays an important role in the field of natural language processing to predict semantic relationships between entities in a sentence. Currently, most models have typically utilized the natural language processing tools to capture high-level features with an attention mechanism to mitigate the adverse effects of noise in sentences for the prediction results. However, in the task of relational classification, these attention mechanisms do not take full advantage of the semantic information of some keywords which have information on relational expressions in the sentences. Therefore, we propose a novel relation extraction model based on the attention mechanism with keywords, named Relation Extraction Based on Keywords Attention (REKA). In particular, the proposed model makes use of bi-directional GRU (Bi-GRU) to reduce computation, obtain the representation of sentences , and extracts prior knowledge of entity pair without any NLP tools. Besides the calculation of the entity-pair similarity, Keywords attention in the REKA model also utilizes a linear-chain conditional random field (CRF) combining entity-pair features, similarity features between entity-pair features, and its hidden vectors, to obtain the attention weight resulting from the marginal distribution of each word. Experiments demonstrate that the proposed approach can utilize keywords incorporating relational expression semantics in sentences without the assistance of any high-level features and achieve better performance than traditional methods.

† Corresponding author: Yuanyuan Zhang (E-mail: 16823650@qq.com; ORCID: 0000-0002-5353-2989).

1. INTRODUCTION

Abundant data on the Web are generated and shared every day, thus the relational facts of subjects (entities) in the text are often utilized to represent the text information to capture associations among those data. Generally, triples are utilized to represent entities and their relations which often indicate unambiguous facts about entities. For example, a triple (e_1, r, e_2) denotes that entity e_1 has a relation r with another entity e_2. Knowledge graphs (KG) such as FreeBase [1] and DBpedia [2] are real examples of such representations in the triple form.

Relation extraction is a sub-task of natural language processing (NLP) that can discover relations between entity pairs and given unstructured text data. Previous work in the area of relation extraction from text heavily depends on kernel and feature methods [3]. Recent research studies utilize data-driven Deep Neural Networks (DNNs) methods to eliminate RE of the conventional NLP approaches since these DNN-based methods [4–6] can automatically learn features instead of manually designed features based on the various NLP tool-kits. Most of them surpassed the traditional methods and achieved excellent results for the RE tasks. Among them, both DNNs-based supervised and distant supervision methods are the most popular and reliable solutions for RE but have their own characteristics. Supervised methods have better performance for the specific domain, while distant supervision methods have better performance for generic domains. Therefore, it is difficult to specify which kind of the above two methods are the best. Hence, the following part introduces the DNN-based supervised methods in detail according to the research of the paper.

According to the structure of DNNs, DNN-based Supervised RE usually is classified into various types such as CNN [6–10], RNN [5, 11, 12], or Mix structure. In addition, some variant RNN networks have been developed in RE systems such as the Long Short Term Memory network (LSTM) [13–15], and Gated Recurrent Unit (GRU) [16]. Each kind of DNN has its own characteristics and advantages in dealing with various language tasks. For example, due to the parallel processing ability, the CNNs are good at addressing local and structural information, but rarely capture global features and time sequence information. Instead, RNNs, LSMTs, and GRUs, which are suitable for modeling sequence and problem transformation, can alleviate these problems that CNNs cannot overcome.

However, these structural RNNs-based methods have a common drawback which is that many external artificial features are introduced without an effective feature filter mechanism [17]. Therefore, the semantic-oriented approaches are utilized to improve the ability of semantic representation via capturing the internal association of text and the attention mechanisms. To alleviate the influence of word-level noise within sentences, many efforts have been devoted to getting rid of irrelevant words [18–21], especially, the recent state-of-the-art attention-based methods such as [19, 22, 23].

Although the inner-sentence noise can be alleviated by the attention mechanisms with the caculation of weights for the each word independently, there are some information for better extraction through some continuous words such as phrases. Yu et al.[24] proposes an attention mechanism based on the conditional random fields (CRF), which incorporates such keywords information into the neural relation extractor.

Compared with other strong feature-based classifiers and all baseline neural models, the CRF mechanism is important for this model to construct a better attention weight.

Based on the above analysis, we propose a novel relation extraction model based on the attention mechanism with keywords, named Relation Extraction Based on Keywords Attention (REKA) , which incorporates an attention mechanism based on the keywords-identifiable of relation that is similar to the segments in the [24]. Different from the model in [24], our model makes use of bi-directional GRU (Bi-GRU) to reduce computation without any NLP tools. In particular, the CRF attention mechanism includes two components: entity pair attention and segment attention.

The proposed entity pair attention means adding additional weight to the entity part of the dataset so that it plays a more decisive role when entering the code. The proposed segment attention is assumed that each sentence has a binary sequence of states corresponding to it and that each state variable in the sequence corresponds to a word in the sentence. This binary state variable indicates whether the corresponding word is related to the relation extraction task with 0 and 1, respectively. Inspired by the [24], we utilized a linear-chain CRF incorporating segment attention to obtain the marginal distribution of each state variable as an attention weight.

To summarize, the contributions of the proposed REKA model are shown as follows:

- Propose a novel Bi-GRU model based on an attention mechanism with keywords to handle the relation extraction.
- Both entity pair similarity features and segment features are incorporated in the proposed attention mechanism with keywords.
- Achieves state-of-the-art performance without any other NLP tools assistance.
- Be more interpretable than the original Bi-GRU model.

2. RELATED WORK

2.1 RNN-Based Relation Extraction Models

Recently, relation extraction research focuses on extracting relational features with neural networks[25–27]. Zhang et al. [28] claimed that RNN-based relation extraction models have better performance than that the CNN-based models since CNN's can only obtain the local features, but RNNs are good at learning long-distance dependency between entities. Afterward, LSTM [15] is proposed by using the gate mechanism to solve the problem of gradient explosion in RNN models. Based on this, Xu et al. [5] propose a model with LSTM via the shortest dependency path (SDP) between entities, named the SDP-LSTM model, in which there are four types of information, including Word vectors, POS tags, Grammatical relations, and WordNet hypernyms, to support external information. To address the problem of shallow architecture difficultly represented by the potential space in different network levels, Xu et al. [29] can obtain the abstract features along the two sub-paths of SDP.

Since dependency trees are directed graphs, it is necessary to identify whether the relation implies the reverse direction or the first entity is related to the second entity. Therefore, the SPD is divided into two sub-paths, each directed from the entity towards the ancestor node. However, one-directional LSTM models lack representation of the complete sequential information. Thus, the bidirectional LSTM model (BiLSTM) is utilized by Zhang et al. [30] to obtain the sentence level representation with several lexical features. The experimental results demonstrate word embedding as an input feature alone is enough to achieve excellent results. However, the SDP can filter the input text but has no extracted features. To address this issue, the attention mechanism is introduced for BiLSTM-based RE[31].

2.2 Attention Mechanisms for Relation Extraction

Since useful information can be presented anywhere in the sentence, some researchers recently have presented attention-based models which can obtain the important semantic information in a sentence.

Zhou et al. [31] propose the attention mechanism in BiLSTM, which automatically got the important features only with the raw text. Similar to the work of Zhou et al. [31], Xiao et al. [32] propose a two-level BiLSTM architecture based on a two-level attention mechanism to extract a high-level representation of the raw sentence.

Although the attention mechanism is used to capture the important features extracted by the model, [31] just presents a random weight without the consideration of prior knowledge. Therefore, EAtt-BiGRU proposed by Qin et al. [33] leverages the entity pair as prior knowledge to form attention weight. Different from Zhou et al.'s [31] work, EAtt-BiGRU applies bi-directional GRU (Bi-GRU) to reduce computation, capture the representation of sentences and adopt a GRU to extract prior knowledge of entity pairs. Zhang et al. [34] propose a Bi-GRU model based on another attention mechanism with the SDP for the prior knowledge, extracting sentence-level features and attention weights. Nguyen et al. [35] have proposed to use a special attention mechanism and introduced dependency analysis that takes into account the interconnections between potential features.

With the proposed BERT model, which has achieved excellent performance on various NLP tasks, more and more studies have started to try to use the BERT model in search matching tasks and achieved very good results. In the latest study on pre-trained models, Wei et al. [36] achieved high metric scores using BERT. Although the BERT model has excellent encoding ability and can fully capture the semantic information of the context in the sentence, it still has problems such as high training costs and long prediction time.

Our model is inspired by Lee et al. [22], but different from the previous works that can only get word-level or sentence-level attention and rarely obtain the degree of correlation between entities and other related words, our model utilizes Bi-GRU instead of BiLSTM to reduce computation. Meaning while, inspired by the attention model designed by Yu et al. [24] for the relation extraction, which is capable of learning phrase-like features and capturing reasonably related segments as relational expressions based on

the CRF, we propose a novel attention mechanism combining the entity pair attention with the segment attention via CRF together.

Although the above methods provide a solid foundation for the research of supervised RE, there are still limitations among them. For example, the insufficient training corpus puzzles the further development of the supervised RE. Therefore, Mintz et al. [37] propose a distant supervision approach strongly based on an assumption in the selection of training examples. Distant supervision methods also achieved excellent results for the RE [38–40]. However, it also has some drawbacks, for example, the noise in the data sets is obvious. Thus, it is difficult to demonstrate which two kinds of above methods are currently the best. Hence, we just research the supervised methods in this paper.

3. METHODOLOGY

The proposed REKA model consists of four components, the structure of which is shown in Figure 1, and the role of each layer is as follows:

- The input layer that contains word vector information and location information.
- The self-attention layer that processes the word vectors to obtain word representations.
- To obtain contextual information about each word in a sentence The Bi-GRU layer is used.
- The keyword-attention layer extracts the key information in the sentence and passes it to the final classification layer.

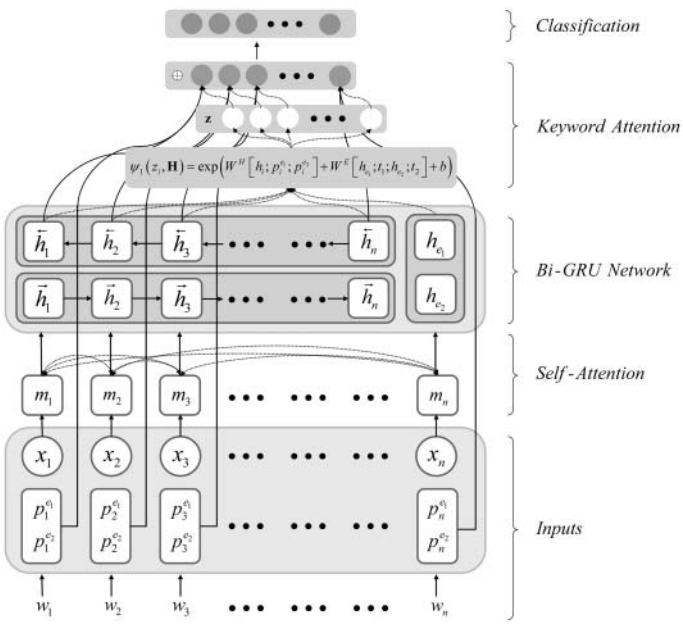

Figure 1. The systematic architecture of the REKA model.

3.1 Input Layer

The REKA model's input layer is designed to transform the original input of the sentence into an embedding vector containing various feature information, where the input sentences are denoted by $\{w_1, w_2, \ldots, w_n\}$ and $\left\{p_1^{e_j}, p_2^{e_j}, \ldots, p_n^{e_j}\right\}$ is a vector of the relative position features information of every word to the entity pair $e_{j\in\{1,2\}}$.

To further enhance the model's ability to better capture the semantic information in sentences, a pre-training model of embedded language models (ELMo) [43] word embedding is utilized in this paper, which proposes a better solution for multiple meanings of words, unlike the previous work of word2vec by Mikolov et al. [41] and GloVe by Pennington et al. [42], in which one word corresponds to a vector that is stationary.

ELMo is a real trained model, in which a sentence or a paragraph is fed into and inferred the word vector corresponding to each word based on the context. One of the obvious benefits of ELMo is that the multiple-meaning words can be understood in the context of the preceding and following words.

After the word embedding process, $\{x_1, x_2, \ldots, x_n\}$ is the d_w dimensional vector and input into the next layer as the position feature vector.

3.2 Multi-Head Attention Layer

Although this paper makes use of non-fixed word vectors in the input layer, we use the Multi-Headed Attention (MHA) mechanism to process the output vectors in the input layer to help the model further understand the deep semantic information in the sentences and to address the problem of long-term dependencies. MHA is a special kind of self-attention mechanism [17, 19], in which the symmetric similarity matrix of the sequences can be constructed from a sequence of word vectors resulting from the input layer.

As shown in Figure 2, given a key **K**, a queries **Q**, and a value **V**, the multi-head attention module will execute the attention h times, the calculation process uses the following equation (1–3):

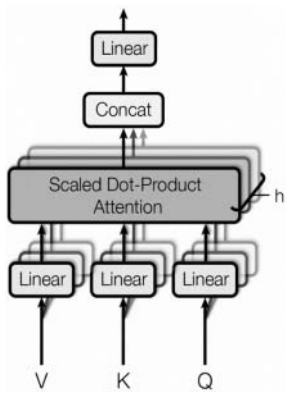

Figure 2. A sample of Multi-Head Attention [17]

$$\text{MultiHead}(Q,K,V) = W^{M}\text{Concat}[\text{head}_1;\ldots;\text{head}_r] \tag{1}$$

$$\text{where head}_i = \text{Attention}\left(W_i^{Q}Q, W_i^{K}K, W_i^{V}V\right) \tag{2}$$

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_w}}\right)V \tag{3}$$

here $W^{M} \in \mathbb{R}^{d_w \times d_w}$, $W_i^{Q} \in \mathbb{R}^{d_w/r \times d_w}$, $W_i^{K} \in \mathbb{R}^{d_w/r \times d_w}$, $W_i^{V} \in \mathbb{R}^{d_w/r \times d_w}$ is the trainable parameter, W^{M} is the scaled dot-product attention calculation when calculated and connected in series, W_i^{Q}, W_i^{K}, W_i^{V} is query, key and value of i^{th} head, respectively [17].

The inputs **Q**, **K**, **V** are all equal to the word embedding vector $\{x_1, x_2, \ldots, x_n\}$ in the multi-head attention[17]. The output of the MHA self-attention is a sequence of features with information about the context of the input sentences.

3.3 Bi-GRU Network

The Bi-GRU network layer was used to obtain semantic information in sentences about the output sequence of the MHA self-attentive layer. As shown in Figure 3, GRU optimizes the LSTM by retaining only two gate operations including a new gate and a reset gate, thus its units, therefore, have fewer parameters and converge faster than LSTM units.

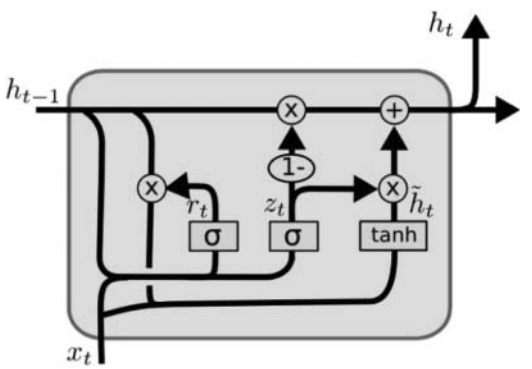

Figure 3. The GRU unit

The GRU unit's processing of m_i is represented in this paper for simplicity as $GRU(m_i)$. Therefore, the equation (4–6) for calculating the contextualized word representation is obtained as follows:

$$\overrightarrow{h_t} = \overrightarrow{GRU}(m_t) \tag{4}$$

$$\overleftarrow{h_t} = \overleftarrow{GRU}(m_t) \tag{5}$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{6}$$

The input ***M*** resulting from the MHA self-attention layer is fed into the Bi-GRU network step by step. To simultaneous use of past and future feature information at a given time step, we connect the hidden state of the forward GRU network $\overrightarrow{h_t} \in \mathbb{R}^d h$ with the hidden state of the backward GRU network $\overleftarrow{h_t} \in \mathbb{R}^d h$ at each step.

Where d_h is used to denote the hidden state of the GRU network unit dimension, $\{h_1, h_2, ..., h_n\}$ is denoted the hidden state vector of each word, The arrow represents the direction of the GRU unit.

3.4 Keywords Attention based on CRF

Although attention mechanisms have achieved state-of-the-art results in a variety of NLP tasks, most of them do not fully exploit the keywords information in the sentences. This is because keywords usually refer to important words for solving relational extraction tasks, and the performance of the models would be improved if information about these keywords could be exploited.

The goal of the attention mechanism with keywords proposed in this paper is to assign more reasonable weights to the hidden layer vectors, where attention weights are also a set of linear combinations of scalars. A more reasonable weight assignment indicates that the model pays more attention to the more important words in the sentence compared to other words, and all the weights in this attention mechanism with keywords take values between 0 and 1.

However, there is a different approach to the calculation of the weights between the traditional attention mechanisms and the proposed model. In particular, the proposed model defines a state variable z for each word in the sentence, it means that the word corresponding to z is irrelevant to the relational classification of this sentence when z equals 0, and vice versa if z equals 1. Thus, each sentence of the input model has a corresponding sequence of z. From the above description, the expected value of a hidden state **N**, the probability of its corresponding word, will be selected and calculated as the following equation (7):

$$\boldsymbol{N} = \sum_i p\left(z_i = 1 | \boldsymbol{H}\right) \boldsymbol{h}_i \tag{7}$$

In order to calculate the $p(z_i = 1|\mathbf{H})$, the CRF is introduced here to calculate the sequence of weights for the hidden sequence vectors $\mathbf{H} = \{h_1, h_2, ..., h_n\}$, where **H** represents the input sequence and h_i represents the hidden output of the GRU layer for the i^{th} word in the sentence. CRF provides a calculation of transfer probabilities for the computation of conditional probabilities in between sequences.

The linear-chain CRF defines a range of conditional probability $p(z_i = 1|\mathbf{H})$ given **H** with the following definition (8–9):

$$p(\boldsymbol{z}|\boldsymbol{H}) = \frac{1}{Z(\boldsymbol{H})} \prod_{c \in C} \psi\left(\boldsymbol{z}_c, \boldsymbol{H}\right) \tag{8}$$

$$Z(\boldsymbol{H}) = \sum_{\boldsymbol{z}' \in Z} \prod_{c \in C} \psi\left(\boldsymbol{z}_c', \boldsymbol{H}\right) \tag{9}$$

Where $\mathcal{Z}$ is the set of state sequences, $Z(\mathbf{H})$ denotes the normalization constant and $\mathbf{Z}_C$ is the subset of z given by individual clique c, $\psi(\mathbf{Z}_C, \mathbf{H})$ is the potential function of this clique. It is defined by the following equation (10):

$$\prod_{c \in C} \psi(\boldsymbol{z}_c, \boldsymbol{H}) = \prod_{i=1}^{n} \psi_1(z_i, \boldsymbol{h}_i) \prod_{i=1}^{n-1} \psi_2(z_i, z_{i+1}) \tag{10}$$

For feature extraction, the feature extractor makes use of two types of feature functions, the vertex feature function $\psi_1(z_i, \mathbf{H})$, the edge feature function $\psi_2(z_i, z_{i+1})$. ψ_1 represents the mapping of the output h of GRU to the state variable z, and ψ_2 simulates the transition of two state variables at adjacent time steps. The equations for their definitions are shown as the following equation (11–13) respectively:

$$\psi_1(z_i, \boldsymbol{H}) = \exp\left(\boldsymbol{W}^H F_1 + \boldsymbol{W}^E F_2 + b\right) \tag{11}$$

$$F_1 = [h_i; p_i^{e_1}; h_{e_2}; t_2];\ F_2 = [h_{e_1}; t_1; h_{e_2}; t_2] \tag{12}$$

$$\psi_2(z_i, z_{i+1}) = \exp\left(\boldsymbol{W}^t_{z_i, z_{i+1}}\right) \tag{13}$$

Where $\mathbf{W}^H$ and $\mathbf{W}^E$ are trainable parameters, b is a trainable bias term. They calculate the contextual information as a feature score for each state variable, which takes advantage of the entity location features $p_i^{e_1} p_i^{e_2}$ as well as keyword features embedded vectors (entity pair hidden similarity features t_1, t_2, and entity pair features h_{e_1}, h_{e_2}).

For the hidden vector output by the words after the Bi-GRU layer, the CRF keyword attention mechanism performs soft selection by assigning higher weights to the words in the sentence that are more relevant to the classification. The processing of the sentence by the CRF keywords attention mechanism is shown in Figure 4, The CRF keyword attention in the figure assigns different weights to each word with an example sentence "*The boy ran into the school cafeteria*". In addition to the two entity words "*boy*" and "*cafeteria*", "*into*" in the sentence was also assigned a higher weight relative to the other words, due to the fact that a is the word associated with the relational classification.

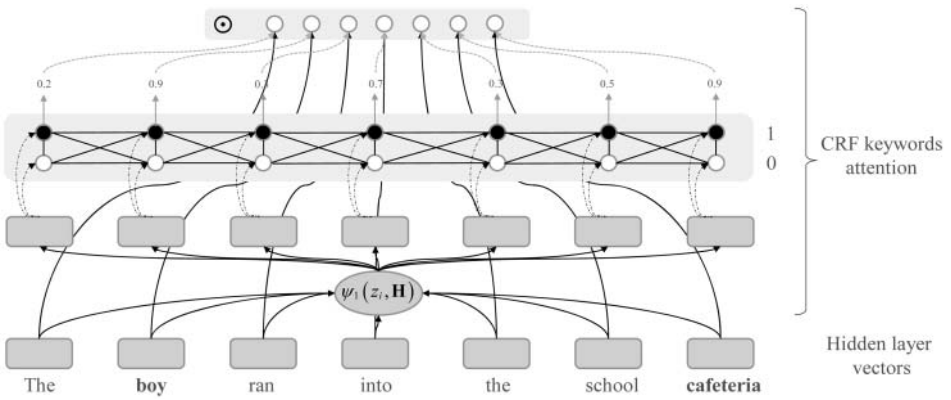

Figure 4. CRF keywords attention mechanism architecture shown with an example sentence "*The boy ran into the school cafeteria*".

Entity position feature: The proposed attention mechanism with keywords in this paper not only obtains word embedding features but also incorporates position embedding features.

In order to represent contextual information as well as the relative location features of entities $p_i^{e_1}, p_i^{e_2}$, this paper connects them with the output of their corresponding hidden layers h_i, as shown by F_1 in Equation 12. There is a definition such as $y_{ctxt} \cdot y_{cand_i}, \ldots, y_{ctxt} \cdot y_{cand_n}$.

Positional vectors are similar to word embedding in that it transforms a relative positional scalar into a feature embedding vector by traversing through the embedding matrix $W_{pos} \in \mathbb{R}^{d_p \times (2L-1)}$, where L is the maximum sentence length, d_p is the dimension of the position vector.

Entity hidden similarity features: Extracting entity hidden similarity features as entity features are used to replace the traditional entity feature extraction method in this paper, thus avoiding the use of traditional NLP tools, and its calculation process is defined as shown in Equation (14–15).

$$a_i^j = \frac{\exp\left(\left(h_{e_j}\right)^\top c_i\right)}{\sum_{k=1}^{K} \exp\left(\left(h_{e_j}\right)^\top c_k\right)} \tag{14}$$

$$t_{j\in\{1,2\}} = \sum_{i=1}^{K} a_i^j c_i \tag{15}$$

In this paper, entities are categorized according to their similarity to their hidden vectors. $c \in \mathbb{R}^{2d_h \times K}$ denotes a potential vector constructed to represent the classes of similar entities, where K is a hyperparameter representing the number of classes in which entities are classified by their hidden similarity.

The j^{th} entity hidden similarity feature t_j is calculated by weighting the similarity of c with the hidden layer output h_{e_j} based on the j^{th} entity.

Entity features are structured by cascading the hidden states corresponding to the entity locations and the potential type representation of the entity pair, shown as F_2 in Equation (12).

3.5 Classification Layer

To compute the probability p of the output distribution of the state variable, A *softmax* layer has been added after the keyword attention layer, which is shown in Equation 16.

$$p(y|\mathbf{N}) = softmax\left(\mathbf{W}_y \mathbf{N} + b_y\right) \tag{16}$$

Of which $|R|$ is the number of relationship categories, $b_y \in \mathbb{R}^{|R|}$ is a biased term, $\mathbf{W}_y$ that maps the expected value of the hidden state $\mathbf{N}$ to the feature score of the relational label.

3.6 Training

The proposed keywords attention is calculated concerning the cross-entropy loss of the relation extraction. This loss function is defined as shown in Equation 17.

$$\mathcal{L}' = -\sum_{i=1}^{|D|} \log p\left(y^{(i)} \left| S^{(i)}, \theta\right.\right) \tag{17}$$

Where $|D|$ is the size of the training data dataset and $(S^{(i)}, y^{(i)})$ is the i^{th} sample in the dataset. The AdaDelta optimizer is utilized to minimize the loss calculation parameter θ in this paper.

To prevent overfitting, L2 regularisation is added to the loss function, where λ_1, λ_2 are the hyperparameters of the regularisation. The second regularizer attempts to compel the model to process a small number of significant words and returns a sparse weight distribution. The resulting objective function $\mathcal{L}$ is shown in Equation 18.

$$\mathcal{L} = \mathcal{L}' + \lambda_1 \|\theta\|_2^2 + \lambda_2 \sum_{i}^{n} p\left(z_i = 1 \left| \boldsymbol{H}\right.\right) \tag{18}$$

4. EXPERIMENTS

4.1 Dataset and Metric

To evaluate the experiment, we used the SemEval-2010 Task 8 dataset for our experiment, SemEval-2010 Task 8 dataset is a benchmark dataset that is widely used in the field of relationship extraction. The dataset has 19 relationship types, including nine directional relationships and others. As shown in Table 1.

Table 1. Types of relationships in the dataset and their percentages.

Type	Number		Rate	
	Training	Testing	Training	Testing
Other	454	1410	17.63	16.71
Cause-Effect	328	1003	12.54	12.07
Component-Whole	312	941	11.76	11.48
Entity-Destination	292	845	10.56	10.75
Product-Producer	261	717	8.96	9.61
Entity-Origin	258	716	8.95	9.50
Member-Collection	233	690	8.63	8.58
Message-Topic	231	634	7.92	8.50
Content-Container	192	540	6.75	7.07
Instrument-Agency	156	504	6.30	5.74

The dataset includes 10717 sentences, of which 8000 samples were used for training and other 2717 samples for testing. The evaluation metrics used here are the macro averaged F1 score based, which is the official evaluation metric of the dataset.

4.2 Implementation Details

In this paper, a publicly available pre-trained EMLo model is used to initialize the word embeddings in the REKA model, and the other weights in the model are initialized randomly using the zero-mean Gaussian distribution, the relevant hyperparameters are shown in Table 2, The grid search was used for the selection of regularised coefficient values for λ_1 and λ_2 from 0 to 0.2.

Table 2. Hyperparameters of our model.

Hyper-parameter	Description	Value
dropout rate	Keyword attention layer	0.5
	Bi-GRU layer	0.6
	Word embedding layer	0.8
	Multi-head attention layer	0.8
λ_1	Regularization coefficient	[0, 0.2]
λ_2		
r	Number of Heads	4
batch size	Size of mini-batch	50
r_1	Initial learning rate	4
d_r	The decay rate of leaming	0.5
d_a	Size of attention layer	50
d_h	Size of hidden layer	512
K	Number of the similar entities' classes	4
d_p	Size of position embeddings	50

4.3 Comparison Models

The proposed REKA model is to be compared with the below benchmark model.

(1) *SVM*: The SVM [44] is a Non-Neural Model, which achieves top results in the SemEval-2010 task, but it uses a lot of handcrafted and computationally intensive features such as WordNet, ProBank, FrameNet, etc.

(2) *MV-RNN*. The MV-RNN [45] is an SDP-based model, SDP is a semantic structural feature in sentences. Models with SDP can be iterated along the shortest dependency path between entities.

(3) *CNN*. The CNN [4] is an end-to-end model on the SemEval-2010 task, which means that the data from the input end is directly obtained from the output end. This model builds a convolutional neural network to learn the feature vector of sentence level.

(4) *BiLSTM*. The BiLSTM [30] is proposed to obtain sentence-level representations on the SemEval-2010 task with bidirectional long short-term memory networks. It is the classic RNN-based relation extraction model.

(5) *DepNN*. The DepNN [46] employs an RNN to model subtrees and a CNN to capture features on the shortest path in sentences.

(6) *FCM*. The FCM [45] decomposes each sentence into sub-structures, then extracts their features separately and finally merges them into the classification layer.

(7) *SDP-LSTM*. The SDP-LSTM [5] employs the long short term memory (LSTM) to capture features along the shortest dependency path (SDP). The model is a convolutional neural network for classification by ranking and uses a loss function with pairwise rank.

(8) *Purely self-attention* [47]. Only a self-attentive coding layer was utilized and combined with a position-aware encoder for relational classification.

(9) *CASREL BERT* [36]. *CASREL BERT* presents a cascade binary tagging framework (CASREL) and implements a new tagging framework that achieves some performance improvements.

(10) *Entity-Aware BERT* [48]. The method builds on BERT with structured predictions and an entity-aware self-attentive layer, achieving excellent performance on the SemEval 2010 Task 8 dataset.

4.4 Experimental Results

To evaluate the proposed models further, we chose the RNN-based model from the above models for comparison. The Precision-Recall (PR) curves and complexity analysis of the models are shown in Figure 5.

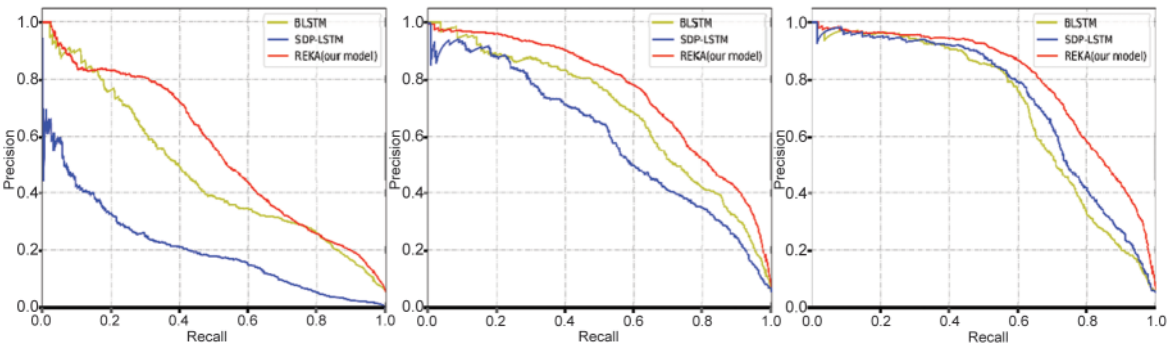

Figure 5. Precision-Recall curves of different used numbers of datasets (1%, 20%, 100%, respectively) and compared with RNN methods.

The comparison results between the REKA model and other models are shown in Table 3, the average precisions (AP) of REKA compared with RNN methods are shown in Table 4.

Table 3. Comparison of the results of the Semeval-2010 Task 8 test dataset.

Model	Additional Features[a]	F1
SVM[42]	POS, WN, etc.	82.3
MV-RNN[43]	POS, NER, WN	82.4
CNN[4]	PE, WN	82.7
BiLSTM[20]	None,	82.7
	+ PF, POS, etc.	84.3
DepNN[44]	DEP	83.6
FCM[45]	SDP, NER	83.0
SDP-LSTM[5]	SDP	83.7
Purely Self-Attention[47]	PE	83.8
CASREL BERT	PE	87.5
Entity-Aware BERT[48]	PE	88.8
REKA Model	PE	84.8

Notes: a. (Where WN, DEP, SDP, PE are WordNet, dependency features, shortest dependency path, position embeddings, respectively).

Table 4. Average precision score for our model and compared methods (micro-averaged over all classes).

[a]	BiLSTM	SDP-LSTM	REKA
1%	0.26	0.47	**0.55**
20%	0.60	0.68	**0.76**
100%	0.73	0.70	**0.81**

Notes: a. (The first columns show how much of testing data has been used. Performance is on the SemEval-2010 task dataset).

The experimental results show that the proposed REKA model is superior to the conventional model with fewer features but is lower than the Entity-Aware BERT and CASREL BERT. However, the pre-trained model file of the BERT is so large that it takes longer to be trained with higher hardware performance requirements.

As shown in Table 5, we conducted ablation experiments on the development dataset in order to explore the contribution of the various components of the keywords-aware attention mechanism to the experimental results. We gradually stripped the individual components from the original model, the experimental results showed that the F1-score decreased by 0.2 when the position embedding component was stripped from the model. MHA, pre-trained EMLo word embeddings, and entity is hidden similarity features provide F1 scores of 0.5, 1.2, and 0.8 respectively for the model. In particular, a 2.3% improvement of F_1 is a result of the keywords-aware attention. Therefore, experimental results demonstrate that these components contribute to the model in a complementary way rather than working individually and achieve an F1 score of 84.6 via the combination of all components.

Table 5. The effect of components on the F_1-score of the model.

Model	Dev F_1
Our model	84.6
- Position embedding	84.4
- Multi-head attention	83.9
- Pre-trained EMLo word embeddings	82.7
- Entity hidden similarity features	81.9
- Keyword-aware attention	79.6

5. CONCLUSION

In this paper, we propose a novel Bi-GRU network model based on an attention mechanism with keywords for the task of RE on the SemEval-2010 task dataset. This model adequately extracts features that are available in the dataset through the keyword attention mechanism and achieved F1 score of 84.8 without the use of other NLP tools. To calculate the marginal distribution for each word, we used the similarity between the output of the hidden vectors by the entity words in the hidden layer and the relative position feature vectors between the entity words in the CRF keyword attention mechanism, which is chosen as the attention weight. Our further research will be carried out on attention mechanisms that can better extract key information from sentences, and we are planning to use this for the identification of relationships between several entities.

ACKNOWLEDGMENTS

This work is supported by the Science and Technology Project of Hubei Electric Power Co., LTD., State Grid (149).2020

AUTHOR CONTRIBUTIONS

Yuanyuan Zhang (E-mail: 16823650@qq.com, ORCID:0000-0002-5353-2989): has participated in the proposed model design and writing of the manuscript.

Yu Chen (E-mail: 1148848330@qq.com, ORCID: 0000-0001-7316-3570): has participated in the coding, the experiment and analysis, writing the manuscript.

Shengkang Yu (E-mail: 12052033@qq.com, ORCID:0000-0001-6374-3395): has participated in the part of the experiment and analysis.

Xiaoqin Gu (E-mail: 1564785699@qq.com, ORCID: 0000-0001-6308-8474): has participated in the part of the experiment and analysis.

Mengqiong Song (E-mail: 297365728@qq.com, ORCID:0000-0002-2816-5670): has participated in the part of the experiment and analysis.

Yu Peng (E-mail: 1039079148@qq.com, ORCID:0000-0002-5353-2989): has participated in the revision of the manuscript.

Jianxia Chen (E-mail: 1607447166@qq.com, ORCID: 0000-0001-6662-1895): has participated in the model design, problem analysis, writing and revision of the manuscript.

Qi Liu (E-mail:260129443@qq.com, ORCID:0000-0003-1066-898X): has participated in the writing and revision of the manuscript.

REFERENCES

[1] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J.: Free-base: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp. 1247–1250 (2008, June)

[2] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z.: Dbpedia: A nucleus for a web of open data. In The semantic web, pp. 722–735 (2007). Springer, Berlin, Heidelberg

[3] Pawar, S., Palshikar, G.K., & Bhattacharyya, P.: Relation extraction: A survey. arXiv preprint arXiv:1712.05191 (2017)

[4] Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J.: Relation classification via convolutional deep neural network. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers, pp. 2335–2344 (2014, August)

[5] Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., & Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1785–1794 (2015, September)

[6] Liu, C., Sun, W., Chao, W., & Che, W.: Convolution neural network for relation extraction. In: International conference on advanced data mining and applications, pp. 231–242 (2013, December). Springer, Berlin, Heidelberg

[7] Nguyen, T.H., & Grishman, R.: Relation extraction: Perspective from convolutional neural networks. In: Proceedings of the 1st workshop on vector space modeling for natural language processing, pp. 39–48 (2015, June)

[8] Santos, C.N.D., Xiang, B., & Zhou, B.: Classifying relations by ranking with convolutional neural networks. arXiv preprint arXiv:1504.06580 (2015)

[9] Chen, Y.: Convolutional neural network for sentence classification (Master's thesis, University of Waterloo) (2015)

[10] Kalchbrenner, N., Grefenstette, E., & Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)

[11] Elman, J.L. Distributed representations, simple recurrent networks, and grammatical structure. Machine learning 7(2), 195–225 (1991)

[12] Zhang, D., & Wang, D.: Relation classification via recurrent neural network. arXiv preprint arXiv:1508.01006 (2015)

[13] Zhang, S., Zheng, D., Hu, X., & Yang, M.: Bidirectional long short-term memory networks for relation classification. In: Proceedings of the 29th Pacific Asia conference on language, information and computation, pp. 73–78 (2015, October)

[14] Sundermeyer, M., Schlüter, R., & Ney, H.: LSTM neural networks for language modeling. In: Thirteenth annual conference of the international speech communication association (2012)

[15] Hochreiter, S., & Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)

[16] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)

[17] Wang, H., Qin, K., Zakari, R. Y., Lu, G., & Yin, J.: Deep neural network-based relation extraction: an overview. Neural Computing and Applications, 1–21 (2022)

[18] Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., & Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1785–1794 (2015, September)

[19] Zhang, Y., Zhong, V., Chen, D., Angeli, G., & Manning, C.D.: Position-aware attention and supervised data improve slot filling. In: Conference on Empirical Methods in Natural Language Processing (2017)

[20] Zhang, Y., Qi, P., & Manning, C.D.: Graph convolution over pruned dependency trees improves relation extraction. arXiv preprint arXiv:1809.10185 (2018)

[21] Liu, T., Zhang, X., Zhou, W., & Jia, W.: Neural relation extraction via inner-sentence noise reduction and transfer learning. arXiv preprint arXiv:1808.06738 (2018)

[22] Lee, J., Seo, S., & Choi, Y.S.: Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. Symmetry 11(6), 785 (2019)

[23] Wang, H., Qin, K., Lu, G., Luo, G., & Liu, G.: Direction-sensitive relation extraction using bi-sdp attention model. Knowledge-Based Systems 198, 105928 (2020)

[24] Yu, B., Zhang, Z., Liu, T., Wang, B., Li, S., & Li, Q.: Beyond Word Attention: Using Segment Attention in Neural Relation Extraction. In: IJCAI, pp. 5401–5407 (2019, August)

[25] Aydar, M., Bozal, O., & Ozbay, F.: Neural relation extraction: a survey. arXiv e-prints, arXiv-2007 (2020)

[26] Socher, R., Huval, B., Manning, C.D., & Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pp. 1201–1211 (2012, July)

[27] Zeng, D., Liu, K., Chen, Y., & Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1753–1762 (2015, September)

[28] Zhang, D., & Wang, D.: Relation classification via recurrent neural network. arXiv preprint arXiv:1508.01006 (2015)

[29] Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., & Jin, Z.: Improved relation classification by deep recurrent neural networks with data augmentation. arXiv preprint arXiv:1601.03651 (2016)

[30] Zhang, S., Zheng, D., Hu, X., & Yang, M.: Bidirectional long short-term memory networks for relation classification. In: Proceedings of the 29th Pacific Asia conference on language, information and computation, pp. 73–78 (2015, October)

[31] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers), pp. 207–212 (2016, August)

[32] Xiao, M., & Liu, C.: Semantic relation classification via hierarchical recurrent neural network with attention. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1254–1263 (2016, December)
[33] Qin, P., Xu, W., & Guo, J.: Designing an adaptive attention mechanism for relation classification. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 4356–4362 (2017, May). IEEE
[34] Zhang, C., Cui, C., Gao, S., Nie, X., Xu, W., Yang, L., ... & Yin, Y.: Multi-gram CNN-based self-attention model for relation classification. IEEE Access 7, 5343–5357 (2018)
[35] Zhang, C., Cui, C., Gao, S., Nie, X., Xu, W., Yang, L., ... & Yin, Y.: Multi-gram CNN-based self-attention model for relation classification. IEEE Access 7, 5343–5357 (2018)
[36] Wei, Z., Su, J., Wang, Y., Tian, Y., & Chang, Y.: A novel cascade binary tagging framework for relational triple extraction. arXiv preprint arXiv:1909.03227 (2019)
[37] Mintz, M., Bills, S., Snow, R., & Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 1003–1011 (2009, August)
[38] He, Y., Li, Z., Yang, Q., Chen, Z., Liu, A., Zhao, L., & Zhou, X.: End-to-end relation extraction based on bootstrapped multi-level distant supervision. World Wide Web 23(5), 2933–2956 (2020)
[39] Han, X., Liu, Z., & Sun, M.: Neural knowledge acquisition via mutual attention between knowledge graph and text. In: Proceedings of the AAAI Conference on Artificial Intelligence 32(1) (2018, April)
[40] Wang, G., Zhang, W., Wang, R., Zhou, Y., Chen, X., Zhang, W., ... & Chen, H.: Label-free distant supervision for relation extraction via knowledge graph embedding. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp. 2246–2255 (2018)
[41] Mikolov, T., Chen, K., Corrado, G., & Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
[42] Pennington, J., Socher, R., & Manning, C.D.: Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543 (2014, October)
[43] Sarzynska-Wawer, J., Wawer, A., Pawlak, A., Szymanowska, J., Stefaniak, I., Jarkiewicz, M., & Okruszek, L.: Detecting formal thought disorder by deep contextualized word representations. Psychiatry Research 304, 114135 (2021)
[44] Rink, B., & Harabagiu, S.: Utd: Classifying semantic relations by combining lexical and semantic resources. In: Proceedings of the 5th international workshop on semantic evaluation, pp. 256–259 (2010, July)
[45] Socher, R., Huval, B., Manning, C.D., & Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pp. 1201–1211 (2012, July)
[46] Liu, Y., Wei, F., Li, S., Ji, H., Zhou, M., & Wang, H.: A dependency-based neural network for relation classification. arXiv preprint arXiv:1507.04646 (2015)
[47] Bilan, I., & Roth, B.: Position-aware self-attention with relative positional encodings for slot filling. arXiv preprint arXiv:1807.03052 (2018)
[48] Wang, H., Tan, M., Yu, M., Chang, S., Wang, D., Xu, K., ... & Potdar, S.: Extracting multiple-relations in one-pass with pre-trained transformers. arXiv preprint arXiv:1902.01030 (2019)

AUTHOR BIOGRAPHY

Yuanyuan Zhang (1979–), male, Ph.D. graduated from Wuhan University, associate professor and senior engineer of Technical Training Center of State Grid Hubei Electric Power Co., Ltd. Research direction: intelligent substation technology, intelligent power grid operation and inspection technology, E-mail: 16823650@qq.com, ORCID: 0000-0002-5353-2989

Yu Chen (1996–), male, graduate student of Hubei University of Technology, research direction: Artificial Intelligent, NLP, E-mail: 1148848330@qq.com, ORCID: 0000-0001-7316-3570

Shengkang Yu (1993–), male, master graduated from Huazhong University of Science and Technology, lecturer and intermediate engineer of Technical Training Center of State Grid Hubei Electric Power Co.,Ltd. Research direction: fault diagnosis of electrical equipment, E-mail: 120520338@qq.com., ORCID: 0000-0001-6374-3395

Xiaoqin Gu (1973–), female, master graduated from Hubei University, lecturer of Technical Training Center of State Grid Hubei Electric Power Co., Ltd., Research direction: power grid operation technology, E-mail: 1564785699@qq.com. ORCID: 0000-0001-6308-8474

Mengqiong Song (1991–), female, master graduated from Wuhan University, intermediate engineer of Technical Training Center of State Grid Hubei Electric Power Co., Ltd. Research direction: power grid operation technology, E-mail: 297365728@qq.com, ORCID: 0000-0002-2816-5670

Yu Peng, female, graduated from Wuhan University. Research direction: grid power electronics, E-mail: 1039079148@qq.com, ORCID: 0000-0002-5353-2989

Jianxia Chen is an associate professor in School of Computer Science at Hubei University of Technology. She obtained her MS at Huazhong University of Science & Technology in China. She has worked as a research fellow on the CCF in China and ACM in USA. Her particular research interests are in knowledge graph and recommendation systems.
ORCID: 0000-0001-6662-1895

Qi Liu, female, graduate student of Hubei University of Technology, research direction: Artificial Intelligent, NLP, E-mail: 260129443@qq.com., ORCID: 0000-0003-1066-898X